UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/734,763 | 12/11/2003 | Rabin Sugumar | 004-8639 | 1964 |

42714 7590 08/23/2006

ZAGORIN O'BRIEN GRAHAM LLP (004)
7600B NORTH CAPITAL OF TEXAS HIGHWAY
SUITE 350
AUSTIN, TX 78731-1191

| EXAMINER | |
|---|---|
| COLEMAN, ERIC | |
| **ART UNIT** | **PAPER NUMBER** |
| 2183 | |

DATE MAILED: 08/23/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/734,763 | SUGUMAR ET AL. | |
| | Examiner | Art Unit | |
| | Eric Coleman | 2183 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *14 June 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☒ Claim(s) *36* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

### *Claim Rejections - 35 USC § 101*

1. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 1-19 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

3. Claims 1-19 are directed to methods for use in a processor. These methods comprise conditions for performing operations. The operations comprise substituting specifier for a greater width source register in claim 1, and other conditions for performing other operations in the claims that are abstract. This operations are abstract as they do not produce any tangible result. The operations for claims 1-19 do not transform any article or physical object to different state or thing or provide any tangible result. (see Diehr, 450 US. At 187,209 USPQ at 8; Benson 409 US at 71-72, 175 USPQ at 676-77; AT&T 172 F.3d at 1358-59, 50 USPQ2d at 1452.

4. Claims 2,8,16 include the a greater width source register is substituted for a lesser width source register is the greater width register onto which the lesser width source register is aliased. The substituting operation does not produce any tangible result and the aliasing of the register also does not produce any tangible result.

5. Claims 3,9,17 include substituting the greater width register includes an indication that a lesser width source register is to replace by a greater width register. The indication setting is setting a bit or bits does not produce any tangible result the interpretation of the meaning of the setting of a bit is abstract not tangible.

6. Claim 4 includes if dependency exists between lesser width producer instruction and a greater width consumer instruction substituting plural instructions for the greater width consumer instruction. This is a condition for performing an operation that does not produce any tangible result.

7. Claims 5,10,18,19 include plural instructions are claimed to merge registers and the merging of registers does not produce any tangible result. Also the mere execution of an instruction with a greater width or substitution of instructions using certain registers whether aliased or not does not necessarily produce any tangible result.

8. Claim 6 includes, determination of if dependency exist between a greater width instruction and a lesser width instruction that are either consumer and producer instructions substituting instructions. The mere substitution of instructions does not produce any tangible result.

9. Claim 7 includes stalling at least one instruction of a fetch group if a dependency exist between an instruction in the fetch group and both and active lesser width producer instruction and an active greater width producer instruction. The mere stalling of an instruction upon meeting a condition does not produce any tangibile result.

10. Claims 11 includes generating a first register mask identifying registers to be modified by lesser width instructions active in a pipeline; and generating a second register mask identifying a second register mask identifying registers to be modified by greater width instruction active in the pipeline. The mere masking bits in registers when a condition is met merely selects certain bits and this does not produce any tangible result.

11. Claim 12 includes the determining if a dependency exists includes comparing a lesser width register specifier of an instruction against the second register mask; and comparing a greater width register specifier of an instruction against the first register mask. This is merely a way to make a determination and does not produce any tangible result.

12. Claim 13 includes determining if a dependency exists includes determining if a greater width instruction in a fetch group modifies a lesser width source register specified by a younger instruction in the same fetch group. This is merely a determination and does not produce any tangible result.

13. Claim 14 includes determining if a dependency exists includes if a lesser width instruction in a fetch group modifies a greater width source register specified by a younger instruction in the same fetch group. This is merely a determination and no tangible result is produced.

14. Claim 15 includes, handling a register conflict between a first instruction specifying a greater width registers and a second register instruction specifying a lesser width source register...substituting for the execution the lesser width source register of the second instruction with a greater width source register specifier. The provides for the substituting of registers which does not provide for any tangible result.

***Claim Rejections - 35 USC § 103***

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims1,15,17, 20-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nair (patent No. 6,195,746).

17. Nair taught the invention substantially as claimed including a data processing ("DP") system comprising: As per claims (1,6,15,20,23,31) Nair taught if a dependency exists one instruction and a result stored in a register of differing register type the register type specifiers were converted to match the type specifier in the instruction (e.g., see fig. 5 and col. 6, line 53-col. 8, line 26 and, col. 9, lines 7-60)[Nair taught the changing of register type specifiers changes the execution unit and associated register. The specifier is allocated where the execution unit are of various types is different register widths (e.g., see col. 3, line 50-col. 4, line 32) and Nair taught some execution units performing single precision operations and others performing double precision operations (e.g., see col. 7, lines 26-52). Nair provided a CAST instruction for dynamically converting register type specifier , looking up the specifier type for the source operand, where when the source register specifier type was different from the target register specifier type, the type target register specifier are converted to match the type specifier of the field of the instruction e.g., see col. 5, lines 1-34).

18. Nair taught that the instructions that converted register specifier types comprised LOAD, STORE and CAST instruction (e.g., see col. 9, lines 21-60). Here data in the

register designated by the source in a load or CAST instruction would have been generated and stored in the source register by another (previous) instruction in the course of processing of instructions. Therefore since Nair provided for substitution of register specifier in any case where the type of the specifiers were different then, it would have been obvious to one of ordinary skill that in the Nair teachings during processing of instructions (e.g., single precision and double precision floating point instructions and fixed point instructions) the LOAD or CAST from between a greater width producer instruction (executed in one execution unit) and a lesser width consumer instruction (executed in another execution unit), Nair would have substituted for execution a greater width source register specifier for a lesser width source register specified by lesser width consumer instruction.

19. Further as to claim 15,20, since the use of instruction with greater width source and destination operands (e.g., double precision) and instructions with lesser width source and destination operands (e.g., single precision) and the Nair system provided register type specifiers for each register including the register type providing information such as register size and replaced register specifiers when the specifier type between source and target do not match(as discussed above) In the Nair teachings a consumer instruction comprises a larger width source register specifier than the producer instruction. One of ordinary skill would have been motivated replace the lesser width source register specifier of the second (i.e., consumer) instruction with the greater width source register specifier so that the instruction would be performed properly with the proper size registers.

20. As per claim 3,17,22 Nair taught substituting the greater width register includes setting an indication that the lesser width source register is to be replaced by the greater width register (e.g., see col. 5, lines 1-67 and col. 6, line 53-col. 7, line 52)[the type indicator indicates to the system when a register is to be replaced when the source and target types (such as sizes) are different].

21. Nair taught when there is a type difference between operands (such as different width operands) if the types are not supported by the architecture an exception should be generated (e.g., see col. 7,lines 2-17). Since the exception in the pipeline would have provided a need to wait for the processing of the exception one of ordinary skill would have been motivated to stall the pipeline at least to ensure the data for processing and processing sequence is correct. This provide for logic (the scope provides for any type of logic including one of software or hardware) for producing an exception which would have prevented the one instruction from being delivered for execution if its dependency depends between an instruction in the fetch group and both an active lesser width produce instruction and a active greater width producer instruction.

22. As per claim 21, Nair taught when there is a type difference between operands (such as different width operands) if the types are not supported by the architecture an exception should be generated (e.g., see col. 7,lines 2-17). Since the exception in the pipeline would have provided a need to wait for the processing of the exception one of

ordinary skill would have been motivated to stall the pipeline at least to ensure the data for processing and processing sequence is correct.

***Claim Rejections - 35 USC § 102***

23. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

24. Claim 4,5,18,19,23 is rejected under 35 U.S.C. 102(b) as being anticipated by Prabu (patent No. 6,463,525).

As to the limitations of claim 4,23 Prabu taught if a dependency exists between a lesser width producer instruction and a greater width consumer instruction, substituting plural instructions [a double precision operation is performed on data stored by a plural single precision operations ] (e.g., see col. 3, lines 8-43).

25. As per claim 5,18,19, Prabhu taught three instruction that together perform the operation of merging single precision registers that are mapped to act as a double precision register and performing double precision operations comprising plural operands on the data (e.g., see col. 4, lines 6-31).

***Claim Rejections - 35 USC § 103***

26. Claims 2,3,6-10,16,23,25,26-30, 31-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nair in view of Prabu (patent No. 6,463,525).

27. Nair taught the invention substantially as claimed including a data processing ("DP") system comprising: As per claims (1,6,15,20,23,31) Nair taught if a dependency exists one instruction and a result stored in a register of differing register type the register type specifiers were converted to match the type specifier in the instruction (e.g., see fig. 5 and col. 6, line 53-col. 8, line 26 and, col. 9, lines 7-60)[Nair taught the changing of register type specifiers changes the execution unit and associated register. The specifier is allocated where the execution unit are of various types is different register widths (e.g., see col. 3, line 50-col. 4, line 32) and Nair taught some execution units performing single precision operations and others performing double precision operations (e.g., see col. 7, lines 26-52). Nair provided a CAST instruction for dynamically converting register type specifier , looking up the specifier type for the source operand, where when the source register specifier type was different from the target register specifier type, the type target register specifier are converted to match the type specifier of the field of the instruction e.g., see col. 5, lines 1-34).

28. Nair taught that the instructions that converted register specifier types comprised LOAD, STORE and CAST instruction (e.g., see col. 9, lines 21-60). Here data in the register designated by the source in a load or CAST instruction would have been generated and stored in the source register by another (previous) instruction in the

course of processing of instructions. Therefore since Nair provided for substitution of register specifier in any case where the type of the specifiers were different then, it would have been obvious to one of ordinary skill that in the Nair teachings during processing of instructions (e.g., single precision and double precision floating point instructions and fixed point instructions) the LOAD or CAST from between a greater width producer instruction (executed in one execution unit) and a lesser width consumer instruction (executed in another execution unit), Nair would have substituted for execution a greater width source register specifier for a lesser width source register specified by lesser width consumer instruction.

29. As to the limitations of claim 6, 25,31,33 Nair did not specifically detail substituting an plural instructions for an instruction when a dependency between instructions Prabu taught if a dependency exists between a lesser width producer instruction and a greater width consumer instruction, substituting plural instructions [a double precision operation is performed on data stored by a plural single precision operations ] (e.g., see col. 3, lines 8-43).

30. It would have been obvious to one of ordinary skill in the DP art to combine the teachings of Nair and Prabhu. Both references were directed to the problems of processing instructions with registers of plurality of widths (e.g., see col.3 lines 8-12 of Prabhu and col. 1, line 46-col. 2, line 18 of Nair). One of ordinary skill would have been motivated to incorporate the Prabhu teachings of substituting plural single precision instructions for a double precision instruction to facilitates efficiently processing of the

double precision instruction when the source operands were stored in single precision registers.

31. As per claim 7,26 Nair taught when there is a type difference between operands (such as different width operands) if the types are not supported by the architecture an exception should be generated (e.g., see col. 7,lines 2-17). Since the exception in the pipeline would have provided a need to wait for the processing of the exception one of ordinary skill would have been motivated to stall the pipeline at least to ensure the data for processing and processing sequence is correct. (As to claim 26) The exception logic would have logic to prevent an instruction from being delivered for execution

32. As per claims 2, 8,9,16, 28,31,33 Nair taught the greater width source register substituted for the lesser width source register is the greater width register onto which the lesser width source register is aliased, and setting an indication (e.g., see col. 5, lines 1-54).

Also, Prabhu taught the greater width source register substituted for the lesser width source register is the greater width register onto which the lesser width source register is aliased, and setting an indication (e.g., see col. 3, lines 15-43).

33. As per claim 3,17,22,27 Nair taught substituting the greater width register includes setting an indication that the lesser width source register is to be replaced by the greater width register (e.g., see col. 5, lines 1-67 and col. 6, line 53-col. 7, line 52)[the type indicator indicates to the system when a register is to be replaced when the source and target types (such as sizes) are different].

34. As per claim 10, Prabhu taught three instructions that together perform the operation of merging single precision registers that are mapped to act as a double precision register and performing double precision operations comprising plural operands on the data (e.g., see col. 4, lines 6-31).

35. As to the further limitations of claim 29, 30,32,34,35 the grouping of the fetching of instructions as a group was well known in the art at the time of the claimed invention and one of ordinary skill would have been motivated to fetch plural instructions used as described above as group at least to speed fetching of instructions versus taking the time to fetech each instruction individually.

36. Claims 11,12,13,14,are rejected under 35 U.S.C. 103(a) as being unpatentable over Nair and Prabhu as applied to claims 6 above, and further in view of Yeager (patent No. 6,216,200)(cited in last office action).

37. Prabu taught if a dependency exists between a lesser width producer instruction and a greater width consumer instruction, substituting plural instructions [a double precision operation is performed on data stored by a plural single precision operations ] (e.g., see col. 3, lines 8-43).

38. Yeager taught (claims 11,12,) generating a first register mask identifying registers to be modified by instructions active in a pipeline and generating a second register mask identifying registers to be modified by greater width instructions active in the pipeline, Comparing the register specifier against a second mask and comparing a register specifier against a first register mask (e.g, see figs. 13a,13b,14,25a,25b).

Considering Yeager teachings of masks for determining dependency and the Prabhu teachings of dependency between single and double precision instructions one of ordinary skill would have been motivated generate register masks and to compare the double and single with register specifiers against masks (e.g., see col. 5, line 35-col. 6, line 34).

39. As to the limitations of claim 13, 14, the fetching of data in groups or blocks was well known in the art at the time of the claimed invention at least because the instructions in the same group or block would be more likely to be to be used with a fetched instruction. Therefore it would have been obvious to one of ordinary skill that at least some instructions in the Nair and Prabhu system comprising greater and lesser width instructions that were tested as to their dependency would have been fetched in a group.

40. It would have been obvious to one of ordinary skill to combine the teachings of Prabhu and Yeager. Both references were directed toward the processing of instructions that had dependencies on other instructions. One of ordinary skill would have been motivated to incorporate the Yeager teachings of masks for comparing register modifiers at least to provide quick comparison of the register masks and providing quick determination of dependencies.

41. Claims 24, is rejected under 35 U.S.C. 103(a) as being unpatentable over Prabhu as applied to claims 23 above, and further in view of Nair.

Nair taught the invention substantially as claimed including a data processing ("DP") system comprising: As per claim (24) Nair taught if a dependency exists one instruction and a result stored in a register of differing register type the register type specifiers were converted to match the type specifier in the instruction (e.g., see fig. 5 and col. 6, line 53-col. 8, line 26 and, col. 9, lines 7-60)[Nair taught the changing of register type specifiers changes the execution unit and associated register. The specifier is allocated where the execution unit are of various types is different register widths (e.g., see col. 3, line 50-col. 4, line 32) and Nair taught some execution units performing single precision operations and others performing double precision operations (e.g., see col. 7, lines 26-52). Nair provided a CAST instruction for dynamically converting register type specifier , looking up the specifier type for the source operand, where when the source register specifier type was different from the target register specifier type, the type target register specifier are converted to match the type specifier of the field of the instruction e.g., see col. 5, lines 1-34).

42. Nair taught that the instructions that converted register specifier types comprised LOAD, STORE and CAST instruction (e.g., see col. 9, lines 21-60). Here data in the register designated by the source in a load or CAST instruction would have been generated and stored in the source register by another (previous) instruction in the course of processing of instructions. Therefore since Nair provided for substitution of register specifier in any case where the type of the specifiers were different then, it would have been obvious to one of ordinary skill that in the Nair teachings during processing of instructions (e.g., single precision and double precision floating point

instructions and fixed point instructions) the LOAD or CAST from between a greater width producer instruction (executed in one execution unit) and a lesser width consumer instruction (executed in another execution unit), Nair would have substituted for execution a greater width source register specifier for a lesser width source register specified by lesser width consumer instruction.

43. It would have been obvious to one of ordinary skill in the DP art to combine the teachings of Nair and Prabhu. Both references were directed to the problems of processing instructions with registers of plurality of widths (e.g., see col.3 lines 8-12 of Prabhu and col. 1, line 46-col. 2, line 18 of Nair). One of ordinary skill would have been motivated to incorporate the Nair teachings of use of register specifiers with types for indicating information such as register size in the operations at least so that the comparision of the register type would be facilitate for determining when conflicting size registers are encounters in the processing of the instruction.

***Allowable Subject Matter***

44. Claim 36 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The rejectioin under 35 U.S.C. 101 are maintained as set forth in the last office action and repeated above. As to these rejections the applicants arguments were not persuasive.

***Response to Arguments***

Applicant's arguments with respect to claims 1-35 have been considered but are moot in view of the new ground(s) of rejection.

***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Meier (patent No. 6,105,129) disclosed a system for converting register data from a first format type to a second format type (e.g., see abstract).

Tremblay (patent No. 5,778,247) disclosed a multi-pipeline microprocessor with data precision mode indicator (e.g., see abstract).

Hatta (patent No. 5,515,520) disclosed a DP system for single and double-precision data (e.g., see abstract).

Pontius (patent No. 6,0129,243) disclosed a floating point processor with operand format precision greater than execution precision (e.g., see abstract).

Putrino (patent No. 5,805,475) disclosed load-store unit and storing single-precision floating point registers in a double-precision architecture (e.g., see abstract).

Elliott (patent No. 6,253,312) disclosed a system for double operand load (e.g., see abstract).

Ashton (patent No. 4,847,802) disclosed s system for identifying precision of an operand in a multiprocessor floating-point processor (e.g., see abstract).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Coleman whose telephone number is (571) 272-4163. The examiner can normally be reached on Monday-Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

EC

ERIC COLEMAN
PRIMARY EXAMINER